



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/656,531	09/07/2000	Tim Armandpour	P3929	2317
24739	7590	09/01/2009	EXAMINER	
CENTRAL COAST PATENT AGENCY, INC			BASEHOAR, ADAM L	
3 HANGAR WAY SUITE D			ART UNIT	PAPER NUMBER
WATSONVILLE, CA 95076			2178	
			MAIL DATE	DELIVERY MODE
			09/01/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	09/656,531	ARMANDPOUR ET AL.	
	Examiner	Art Unit	
	ADAM L. BASEHOAR	2178	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 16 June 2009.
 2a) This action is FINAL. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-28 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1-28 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date. _____ .
3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)	5) <input type="checkbox"/> Notice of Informal Patent Application
Paper No(s)/Mail Date <u>08/25/09</u> .	6) <input type="checkbox"/> Other: _____ .

DETAILED ACTION

1. This action is responsive to communications: The Request for Continued Examination (RCE) filed 06/16/09.
2. The objection to claim 18 has been withdrawn as necessitated by the Amendment.
3. The rejection of claims 1-11 under 35 U.S.C. 101 has been withdrawn as necessitated by the Amendment.
4. Claims 1-28 are pending in the case. Claims 1, 12, and 18 are independent claims.

Information Disclosure Statement

5. The information disclosure statement (IDS) submitted on 08/25/09 has been considered by the examiner. The information disclosure statement filed 08/25/09 fails to comply with 37 CFR 1.98(a)(2), which requires a legible copy of each cited foreign patent document; each non-patent literature publication or that portion which caused it to be listed; and all other information or that portion which caused it to be listed. Please note the annotated information disclosure statement which highlights the references that are lacking an appropriately submitted copy.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. Claims 1-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over DaCosta et al (US-6,826,553 11/30/04) in view of Weinberg et al (US-6,360,332 03/19/02) in further view of Heninger (US-6,029,207 02/22/00).

-In regard to substantially similar independent claims 1 and 12, DaCosta teaches an application for enabling automated notification of applied structural changes to electronic information pages on a network comprising:

an interface for enabling users to build and modify network navigation and interaction templates using functional logic blocks for automatically navigating to and interacting with interactive electronic information pages on the network (column 2, lines 11-30 & 55-67; column 3, lines 8-13, 35-43, & 53-65; column 5, lines 30-67; column 7, lines 17-54)(Figs. 1 & 7);

a navigation interface for integrating the software application to a proxy-navigation system for periodic execution of the templates (column 5, lines 19-20: “automatically repeat these steps in a scheduled manner or when requested”);

a change notification module for indicating a navigation and interaction routine has failed and for creating a data file associated with the failed routine (column 18, lines 43-67: “it is known the script has failed...and proper notifications sent to individuals or entities responsible for the operation of the failing script by email...for example”; column 19, lines 1-15); and

sending proper notifications of the failed script to the developer upon failure of the script (column 6, lines 9-13 & 35-41; column 18, lines 53-67; column 19, lines 1-15). DaCosta does not specifically teach storing the data file in a data repository with a point-of-failure indication, parameters associated with the failed routine, and an identifier of the associated electronic information page subjected to the navigation. Weinberg teaches storing the data file (column 2,

lines 39-40; column 6, lines 19-22), wherein the application periodically submits test navigation and interaction routines (column 6, lines 19-22), and upon failure of the routine, creates a data file (column 2, lines 39-40; column 3, lines 29-43; column 6, lines 19-22; column 17, lines 10-52)(Fig. 5F), the data file comprising a point-of-failure indication within the failed routine identifying the logic block of the template that failed (Fig. 5F: column 17, lines 17-21), parameters of the failure (column 17, lines 35-43), an identifier of the associated electronic page (columns 17-18: lines 62-12)(Fig. 5F: “URL: www.mercint.com”), and stores the data file in the data repository sending notification of the action to the developer (column 2, lines 39-40; column 6, lines 15-23). It would have been obvious to one of ordinary skill in the art at the time of the invention to have stored the failed navigation script of DaCosta and for the proper notifications of the failed script to have included a point in process of the failure along with the an identifier of the associated web page, because Weinberg teaches that by storing the failed navigation script, a developer can easily display the results of the navigation and quickly determine the location of the failure of the routine (column 3, lines 29-44). This would have made the re-teaching (i.e. correcting) of the navigation script easier for the developer (column 6, lines 9-13 & 35-41; column 18, lines 42-67).

DaCosta teaches wherein functional logic blocks were part of the navigation and interaction templates containing all of the possible navigation and interaction instructions required by the navigation system-interface module as defined by the a given user/developer (column 2, lines 20-31: “scripts...that locates and extracts data...precisely locating and extracting the select data with a granularity specified by the user” & lines 57-67: “capability for a user to specify...in an automated manor”; column 5, lines 39-55: “learn and store navigation

paths...dialogs and forms that need to be filled...login name and password"; column 7, lines 16-28: "captures each user-generated event."; columns 7-8, lines 55-5: "automatically repeatedly query a web site...upon a single exemplary query"; column 9, lines 5-44). Neither DaCosta nor Weinberg specifically teach wherein the defined functional logic blocks in the defined interaction scripts were modular parts of the interaction scripts. Heninger teaches building software components in a modular fashion such that each modular component could be constructed, modified, and tested independently (column 1, lines 20-29). It would have been obvious to one of ordinary skill in the art at the time of the invention for the functional logic blocks of DaCosta to have been modular parts of the navigation and interaction templates, because Heninger taught that computer software developers realize that modular interacting software components provide the advantages of being more easily designed, generated, tested, installed, and maintained as well leading to better computer products at a minimal cost (column 1, lines 20-67; column 2, lines 1-24). Thus the modular software components of Heninger would have provided the developers of DaCosta a better way of maintaining, editing, and correcting failed navigation scripts (column 18, lines 34-67) by allowing the developers to fix only the modular part of the failed navigation and interaction script.

-In regard to dependent claims 2, 13, and 19, DaCosta teaches wherein the network could be the Internet (column 2, line 13: "Internet") and wherein the electronic information page was a web page (column 2, line 13: "web site") on the network.

-In regard to dependent claim 3, DaCosta teaches wherein the logic blocks include site logic blocks, automated site-login blocks, and automated site-registration blocks (column 2, lines 55-67; column 5, lines 37-43).

-In regard to dependent claim 4, DaCosta teaches wherein the software application was an Internet based application executing and running on a server (column 16, lines 10-50: “smart servers and smart clients...preferably installed...device of the user...download an installer file”; column 18, lines 33-41: “scripts are stored at a central repository that is accessible through the Internet”; column 25, lines 16-21).

-In regard to dependent claim 5, DaCosta teaches wherein the application was accessible through a network browser (column 2, lines 10-30: “Browser”).

-In regard to dependent claim 6, DaCosta teaches wherein the templates are test routines executed for determining success or failure of the routine (column 6, lines 9-13 & 35-41; column 18, lines 54-65).

-In regard to dependent claim 7, DaCosta teaches wherein the templates are executable instruction orders containing logic blocks (column 2, lines 55-67; column 5, lines 37-55; column 6, lines 57-60; column 7, lines 18-54).

-In regard to dependent claim 8, DaCosta teaches wherein the functional logic blocks are modular and self-installable within the templates (column 2, lines 55-67; column 5, lines 37-55; column 6, lines 57-60; column 7, lines 18-54: “stored in an Extensible Markup Language file...programmatically modify the recorded path”)(Fig. 2: 60, 70, 80, 90).

-In regard to dependent claim 9, DaCosta teaches wherein the data files are human readable and are accessed by developers for the purpose of affecting updating of the navigation templates (column 7, lines 29-54; column 18, lines 54-67).

-In regard to dependent claim 10, DaCosta teaches wherein the developers access the application via individual computerized workstations (column 18, lines 34-67)(Fig. 7: “User Developer”).

-In regard to dependent claim 11, DaCosta teaches wherein the error notification and data file are performed in the event failure or a client’s personalized navigation template (column 6, lines 9-13 & 35-41; column 18, lines 34-67).

-In regard to dependent claim 14, DaCosta teaches wherein the software application was an Internet (column 2, line 13: “Internet”) based application executing and running on a server (column 18, lines 26-40).

-In regard to dependent claims 15 and 16, DaCosta teaches wherein a single server system hosting both the proxy navigation software and the software application (column 18, lines 26-40).

-In regard to dependent claim 17, DaCosta teaches wherein software application and the proxy navigation software are integrated as a single application enabling both functions of navigation according to navigation templates and notifying and recoding failed instances of navigation (column 18, lines 26-67).

-In regard to independent claim 18, DaCosta teaches a method for receiving automated notification of random structural changes applied to electronic information pages hosted on a network comprising:

-establishing notification of a failed navigation and interaction routine executed for the purpose of navigating to and interacting with an electronic information page (column 6, lines 9-13 & 35-41; column 18, lines 34-67: “email or pager notification”).

-creating an instance of the failed routine associated with the cause of failure (column 18, lines 43-67: “it is known the script has failed...and proper notifications sent to individuals or entities responsible for the operation of the failing script by email...for example”; column 19, lines 1-15);

-accessing the notification of the failed routine for review purposes (column 6, lines 9-13 & 35-41; column 18, lines 34-67: i.e. developer accesses failed script for re-teaching purposes);

-being able to navigate to the electronic information page identified in the recorded instance (column 6, lines 9-13 & 35-41; column 18, lines 34-67: i.e. developer accesses failed script for re-teaching purposes);

-accessing source information associated with the electronic information page identified in the recorded instance (i.e. re-teaching a new navigation and extraction script by accessing the source information).

-creating new logic block according to the source information and according to information contained in the recorded instance (column 6, lines 9-13 & 35-41; column 18, lines 34-67);

installing the new logic block into existing navigation templates that depend on the updated information for successful function (column 6, lines 9-13 & 35-41; column 18, lines 34-67; column 19, lines 1-15).

DaCosta does not specifically teach wherein the instance of the failed navigation routine was stored for future review including parameters associated with the failed routine that included identification of at least one of a plurality of logic blocks used to build the navigation template. Weinberg teaches storing the data file (column 2, lines 39-40; column 6, lines 19-22), wherein the application periodically submits test navigation and interaction routines (column 6, lines 19-22), and upon failure of the routine, creates a data file (column 2, lines 39-40; column 3, lines 29-43; column 6, lines 19-22; column 17, lines 10-52)(Fig. 5F), the data file comprising a point-

of-failure indication within the failed routine and identifying the logic block of the template that failed (Fig. 5F: column 17, lines 17-21), parameters of the failure (column 17, lines 35-43), an identifier of the associated electronic page (columns 17-18: lines 62-12)(Fig. 5F: “URL: www.mercint.com”), and stores the data file in the data repository sending notification of the action to the developer (column 2, lines 39-40; column 6, lines 15-23). It would have been obvious to one of ordinary skill in the art at the time of the invention to have stored the failed navigation script of DaCosta and for the proper notifications of the failed script to have included a point in process of the failure along with the an identifier of the associated web page, because Weinberg teaches that by storing the failed navigation script, a developer can easily display the results of the navigation and quickly determine the location of the failure of the routine (column 3, lines 29-44). This would have made the re-teaching (i.e. correcting) of the navigation script easier for the developer (column 6, lines 9-13 & 35-41; column 18, lines 42-67).

DaCosta teaches wherein functional logic blocks were part of the navigation and interaction templates containing all of the possible navigation and interaction instructions required by the navigation system-interface module as defined by the a given user/developer (column 2, lines 20-31: “scripts...that locates and extracts data...precisely locating and extracting the select data with a granularity specified by the user” & lines 57-67: “capability for a user to specify...in an automated manor”; column 5, lines 39-55: “learn and store navigation paths...dialogs and forms that need to be filled...login name and password”; column 7, lines 16-28: “captures each user-generated event.”; columns 7-8, lines 55-5: “automatically repeatedly query a web site...upon a single exemplomatic query”; column 9, lines 5-44). Neither DaCosta nor Weinberg specifically teach wherein the functional logic blocks in the defined interaction

scripts were modular parts of the interaction scripts. Heninger teaches building software components in a modular fashion such that each modular component could be constructed, modified, and tested independently (column 1, lines 20-29). It would have been obvious to one of ordinary skill in the art at the time of the invention for the functional logic blocks of DaCosta to have been modular parts of the navigation and interaction templates, because Heninger taught that computer software developers realize that modular interacting software components provide the advantages of being more easily designed, generated, tested, installed, and maintained as well leading to better computer products at a minimal cost (column 1, lines 20-67; column 2, lines 1-24). Thus the modular software components of Heninger would have provided the developers of DaCosta a better way of maintaining, editing, and correcting failed navigation scripts (column 18, lines 34-67) by allowing the developers to fix only the modular part of the failed navigation and interaction script.

-In regard to dependent claim 20, DaCosta teaches wherein the navigation routine was performed according to a test navigation template (Fig. 2: i.e. according to the navigation and extraction scripts).

-In regard to dependent claim 21, DaCosta teaches wherein the navigation routine was performed according to a client navigation template (Fig. 7: “User”).

-In regard to dependent claim 22, DaCosta teaches wherein the recorded instance of the failed routine was created in the form of a data file and stored in a data repository (column 18, lines 54-67).

-In regard to dependent claim 23, DaCosta teaches wherein the recorded instance of the failed navigation routine was accessed by a software developer (column 6, lines 9-13 & 35-41; column 18, lines 54-67).

-In regard to dependent claim 24, DaCosta teaches wherein navigation was performed by the developer utilizing an instance of a browser installed on a computerized workstation (column 2, lines 11-30).

-In regard to dependent claim 25, DaCosta teaches wherein the new logic was in the form of a modular logic block installable to a navigation template (column 6, lines 9-13 & 35-41; column 18, lines 54-67).

-In regard to dependent claim 26, DaCosta teaches wherein the new logic block self-installs to a depended navigation template (column 6, lines 9-13 & 35-41; column 18, lines 42-67: “ensure each of the users has a corrected script as soon as possible, i.e., as soon as it is downloaded to the central repository...running the script”).

-In regard to dependent claim 27, DaCosta teaches testing the new logic before the implementation (column 19, lines 1-15: “determine whether it is operating correctly”).

-In regard to dependent claim 28, DaCosta teaches creating more than one logic block within a navigation template and wherein more than one block could fail (column 6, lines 9-13 & 35-41; column 18, lines 34-67; column 19, lines 1-15).

Response to Arguments

8. Applicant's arguments filed 06/19/09 have been fully considered but they are not persuasive.

-In general, the Applicant argues that it is not clear what portions of the priority continuation-in-part (CIP) application document 09/465,028 are relied upon for providing support in view of the rejection with regard to the DaCosta 6,826,553 patent. The Examiner notes that the disclosure of the CIP is believed to fully support the relied upon/cited sections of the disclosure of the DaCosta reference utilized to reject the claimed subject matter. The Examiner respectfully disagrees with the Applicant's assessment that the teachings in the CIP document 09/465,028 are drastically different from the teachings in the DaCosta patent 6,826,553. As shown in the following/below paragraph, the Examiner has shown specifically in the CIP application where said features are taught and/or supported. By making reference to the DaCosta patent 6,826,553 in rejecting the claims above, the Examiner is making a clear statement that all the rejected claimed elements are taught in the priority document(s).

In regard to the DaCosta reference, Applicant argues for further clarity in view of the priority document(s). Relevant portions of the priority continuation-in-part (CIP) application document 09/465,028 are cited as follows: *(Page 1: “gleaning relevant information from individual web pages...consolidation of data from multiple sources...login names, password, profiles”;* *Page 2: “data location and extraction tool capable of automated operation...displaying the processed data in an organized format...recording a sequence of action...identifying a target pattern”;* *Page 4: “one can automatically...extract specified information based on taught schemas...automatically repeat these steps in a schedule manner or when requested”;* *Page 5: “lean and store navigation paths...dialogs and forms...login name and password...extraction API...records and plays across elements within a single page...pattern matching”;* *Page 6: “data extraction rules...update them separately...data from different sites can be gathered for simultaneous display...according to each user’s particular needs...extraction of statistics, computations, etc”;* *Page 7: “a user interacts...clicking on or activating links or buttons, entering data and so on a is well known...traps each user generated event...forms were submitted...stored in...XML”;* *Page 8: “preferably contains tags indicating the navigation steps and parameters entered on forms”;* *Page 9: “navigation playback...particular page elements”;* *Page 10: “navigation files are preferable encrypted since they may contain login and password information...going to a starting page, performing a series of actions...cause a target or final page to be loaded...accepts as input text selections within an HTML page”;* *Page 11: “three different techniques can be used for identifying a pattern...quick access to relevant information”;* *Page 12: “output of recording module...applied to another page with similar structure”;* *Page 13: “extracted...text information, but also pictures, charts,*

etc."; *Page 14*: "*user needs only to start the recording module...user can utilize...find a particular string...account type, symbol, description...market value*"; *Page 20*: "*XML step is a file...results in a subroutine (or function) being added*"; *Page 21*: "*to perform automated navigations and extractions*"; *Page 23*: "*promote a developer community...for downloading*"; *Page 24*: "*scripts might be checked by the provider for potential errors...include new scripts...scripts are stored at a centralized repository...multiple end users of a script...corrected script as soon as possible...some way to audit or confirm that navigation and extraction scripts are still operational*"; *Page 25*: "*script automatically disabled, and proper notifications sent to individuals or entities responsible for the operation of the failing script by email...less frequently*) (*Figs. 7 & 11*).

In general, it is noted that any citation to specific, pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way. A reference is relevant for all it contains and may be relied upon for all that it would have reasonably suggested to one having ordinary skill in the art. *In re Heck*, 699 F.2d 1331, 1332-33,216 USPQ 1038, 1039 (Fed. Cir. 1983) (quoting *In re Lemelson*, 397 F.2d 1006, 1009, 158 USPQ 275, 277 (CCPA 1968)).

The Examiner notes MPEP § 2144.01, that quotes *In re Preda*, 401 F.2d 825,159 USPQ 342, 344 (CCPA 1968) as stating "in considering the disclosure of a reference, it is proper to take into account not only specific teachings of the reference but also the inferences which one skilled in the art would reasonably be expected to draw therefrom." Further MPEP 2123, states that "a reference may be relied upon for all that it would have reasonably suggested to one

having ordinary skill the art, including nonpreferred embodiments. Merck & Co. v. Biocraft Laboratories, 874 F.2d 804, 10 USPQ2d 1843 (Fed. Cir.), cert. denied, 493 U.S. 975 (1989).

-Applicant also argues that DaCosta's navigation and extraction modules cannot read on Applicant's site logic blocks. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references. As shown above in the rejection, the DaCosta reference specifically teaches wherein a user can define a plurality of different navigation and extraction modules, each having a plurality of different steps, which can access multiple different web sites/pages to automatically extract any number and/or type of relevant information from said web pages for said user. The recorded navigation and extraction steps included stored functional logic for: "dialogs", "forms", "login name and password", "preferences", "URL", "pattern matching", "learning when presented with dynamically generated web pages"; "activating links or buttons"; "entering data and so on as is well known"; "clicks and keyboard input"; "hyperlinks and form fields that are acted upon"; "going to a starting page...submitting forms"; "input text selections"; "three different techniques...identifying a pattern", "highlights the desired information", etc. Each one of said recorded steps is equivalent to the claimed functional logic blocks, wherein each step has an identifiable interaction task with a given element of any number of a plurality of web pages. Thus the totality of the different navigation and extraction steps makes up the claimed navigation and interaction templates.

-In regard to the independent claims Applicant generally argues that Weinberg does not teach the elements of the claimed database interface module such as indicating a point in process

where a navigation routine and interaction routine has failed and for creating a data file containing parameters associated with the failed routine.

Weinberg clearly teaches storing the a data file (column 2, lines 39-40; column 6, lines 19-22), wherein the an application periodically submits test navigation and interaction routines (column 6, lines 19-22), and upon failure of the routine, creates a data file (column 2, lines 39-40; column 3, lines 29-43; column 6, lines 19-22; column 17, lines 10-52)(Fig. 5F), the data file comprising a point-of-failure indication within the failed routine (Fig. 5F: column 17, lines 17-21), an identifier of the associated electronic page (columns 17-18: lines 62-12)(Fig. 5F: “URL: www.mercint.com”), and stores the data file in the data repository sending notification of the action to the developer (column 2, lines 39-40; column 6, lines 15-23). In view of the drawings, Weinberg also clearly teaches recording a point-of-failure indication (Fig. 5F: 88 & 89) within the failed routine, indicating that that verification step failed and thus the status of the test as a whole had failed (column 17, lines 50-52). As discussed before, Weinberg teaches wherein results of the test navigation and interaction routines, including the results of the verification steps were stored for viewing (column 2, lines 39-40). Weinberg also teaches wherein displaying the test results in a hierarchical tree (“report tree”) can also display the results of the verification steps graphically within the report tree, such as displaying a green check mark or a red “X” symbol to indicate pass/fail status (column 3, lines 29-43; column 17, lines 10-52). Thus the Weinberg reference indicates to the developer via the report tree the point-in-process has failed by displaying a red “X” symbol in the report tree (Fig. 5F: i.e. Red “X” shows that Test Iteration 4 has failed. The Test Status (90) also shows that the current test status is “Failed”).

In general, the Examiner respectfully disagrees and believes that the testing tool of Weinberg, which records interactions and navigations between a web browser and web server (column 2, lines 23-40) and then reports the location of the failures of the repeatedly run routines, meets all the claimed limitations to which the reference has been relied upon.

-In regard to pages 19-20 of Applicant's Remarks section, the Applicant generally alleges that the combination of the cited reference still fail to teach or suggest functional logic blocks in defined interaction scripts as well as upon failure of a test routine, creating a data file comprising a point in failure indication within the failed routine identifying the functional logic block in the associated template. As shown above, these features are believed to be clearly taught by the cited prior art references. The Examiner notes that the claimed functional/site-logic blocks have been given their broadest reasonable interpretation in view of the prior art and as such said limitations are clearly taught by said references. DaCosta clearly teaches recording functional navigation and extraction steps that make up specific navigation and interaction scripts. Each of the steps are recorded/stored in an XML file for future playback access as well as future modification. Additionally, Weinberg also teaches a system for automatically recording a series of functional user navigation and interaction steps between a user and a transaction server. Weinberg further taught that upon future playback of said recorded series of steps, said system could record/indicate a point-in-process where a given step in the recorded sequence of steps failed.

Conclusion

9. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Please note the additionally cited references on the accompanying PTO-892 Form.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Adam L. Basehoar whose telephone number is (571)-272-4121. The examiner can normally be reached on M-F: 7:00am - 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Steve Hong can be reached on (571) 272-4124. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Adam L Basehoar/
Primary Examiner, Art Unit 2178